UNITED STATES PATENT APPLICATION FOR:

METHOD AND SYSTEM FOR COMPLETING A BACKUP JOB THAT WAS INTERRUPTED DURING A BACKUP PROCESS

INVENTORS:

GREGG HAMILTON

ASHUTOSH BAHADURE

SACHIN BOBADE

ATTORNEY DOCKET NUMBER: VRTS/0441

CERTIFICATION OF MAILING UNDER 37 C.F.R. 1.10

I hereby certify that this New Application and the documents referred to as enclosed therein are being deposited with the United States Postal Service on Sept. 8, 2003, in an envelope marked as "Express Mail United States Postal Service", Mailing Label No. EV360627484US, addressed to: Mail Stop PATENT APPLICATION, Commissioner for Patents, P.O. Box 1450, Alexandria, VA. 22313-1450.

Signature

Kathleen Faughnan
Name

Sept. 8, 2003
Date of signature

METHOD AND SYSTEM FOR COMPLETING A BACKUP JOB THAT WAS INTERRUPTED DURING A BACKUP PROCESS

BACKGROUND OF THE INVENTION

Field of the Invention

[0001] Embodiments of the present invention generally relate to computer systems, and more particularly, to software for backing up data for computer systems.

Description of the Related Art

[0002] Backing up data for computer systems generally involves making a copy of that data, e.g., creating copies of that data in a database, another computer, disk, tape, and the like. The circumstances under which data is backed up are generally referred to as a session, a job or an event.

[0003] The backup services are generally performed by a backup system, such as computer, a server cluster or a plurality of clusters. The backup system may fail due to a number of operational faults, such as disk failures, and environmental faults, such as power outages caused by natural disasters. When the backup system fails during a backup process, the backup job is interrupted, thereby rendering the backup job incomplete. Typically, once the backup system is reactivated after the failure, the interrupted backup job would have to be processed again from the beginning, which increases the completion time for the entire backup process and the amount of resources needed to complete the job.

[0004] Therefore, a need exists in the art for a method and system for completing a backup job that was interrupted during the backup process from the point of failure rather than from the beginning of the backup job.

SUMMARY OF THE INVENTION

[0005] Embodiments of the present invention are generally directed to a method for completing a backup job that was interrupted during a backup process. After the backup server service identifies the interrupted job, the job manager builds a list of volumes associated with the interrupted job and removes from that list a set of volumes that correspond to the persistent record of completed volumes.

[0006] The job manager along with the catalog manager then identifies a volume that had been partially backed up during the interruption. In one embodiment, the partially backed up volume may be identified using the persistent records of the temporary catalog files and the amount data (bytes) that have been written to the storage devices. Once the partially backed up volume has been identified, the catalog server generates a disk-based catalog containing the partially backed up volume.

[0007] Prior to connecting to the data server to begin the process of writing data stored in the client computer to the storage devices, the job manager initializes the media server and the catalog server. Once the media server and the catalog server are initialized, the data server makes a determination as to whether each container object (directory) in the client computer is listed in the disk-based catalog. If the container object is not listed in the disk-based catalog, then the data associated with the container object are written to the storage devices. If the container object is listed, then the data server further determines whether the container object was partially backed up or completely backed up. If the data server determines that the container object is partially backed up, then the data associated with the container object are written to the storage devices. If the data server determines that the container object is completely backed up, then the data associated with the container object are skipped from being written to the storage devices.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The following detailed description makes reference to the accompanying drawings, which are now briefly described.

[0009] Figure 1 illustrates a block diagram of a computer network that operates in accordance with one embodiment of the present invention.

[0010] Figure 2 illustrates a relational view between the backup system and one of the client computers in accordance with one embodiment of the invention.

[0011] Figure 3 illustrates a flowchart of a method for processing a backup job that was interrupted during a back up process in accordance with one embodiment of the invention.

[0012] While the invention is described herein by way of example for several embodiments and illustrative drawings, those skilled in the art will recognize that the invention is not limited to the embodiments or drawings described. It should be understood, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the present invention as defined by the appended claims. The headings used herein are for organizational purposes only and are not meant to be used to limit the scope of the description or the claims. As used throughout this application, the word "may" is used in a permissive sense (i.e., meaning having the potential to), rather than the mandatory sense (i.e., meaning must). Similarly, the words "include", "including", and "includes" mean including, but not limited to.

DETAILED DESCRIPTION

[0013] Figure 1 illustrates a block diagram of a computer network 100 in which embodiments of the invention may be utilized. The computer network 100 comprises a plurality of client computers 102_1, 102_2, ...102_n that are connected to a backup system 106 via a communications network 104. The backup system 106 may be a single computer, a server cluster or a plurality of server clusters. The backup system 106 is generally configured to provide backup services to the client computers 102. Backup services generally include providing backup for data stored on the client computers 102 and restoration of the data if the original data are lost or corrupted.

[0014] The client computers 102_1, 102_2,...102_n may contain one or more individual computers, workstations, wireless devices, personal digital assistants, desktop computers, laptop computers or any other digital device that may benefit from connection to the computer network 100. Each client computer 102 generally comprises a central processing unit (CPU), support circuits, and memory. The support circuits are well known circuits used to promote functionality of the CPU. Such circuits may include cache, power supplies, clock circuits, input/output interface circuits, and the like. The memory may comprise one or more of random access memory, read only memory, flash memory, removable disk storage, and the like. The memory may store various software packages, such as an operating system software.

[0015] The communication network 104 may be one of many types of networks such as a local area network, wide area network, wireless network, or combinations thereof.

[0016] The backup system 106 is further connected to one or more storage devices 150, which are configured to store data from the client computers 102. Such data may include objects, such as files (leaf objects) and directories (container objects) of drives on the client computers 102. The storage devices 150 may be tape drives, DASD, and the like. Although the storage devices 150 are depicted as being outside of the backup system 106, the storage devices 150 may also be stored inside the backup system 106.

[0017] Figure 2 illustrates a relational view between the backup system 106 and one of the client computers 102 in accordance with one embodiment of the invention. The backup system 106 includes a backup server service 210 generally configured to receive user requests to create new jobs (e.g., backup, restore, etc), to store jobs in a job database 215 and to submit jobs to the job manager 230 for execution at their scheduled times. The backup server service 210 may also be configured to maintain statistics on both active and completed jobs, to provide the user with a means for accessing these statistics, and to provide the user with a means for configuring the backup system 106 and the storage devices 150. Each backup job generally includes information regarding the volumes to be backed up and the scheduled run time, all of which are typically stored in the job database 215. As such, the job database 215 contains a list of jobs that need to be run with their scheduled run times, as well as historical information about the jobs, such as, job run times, statistical information and the like.

[0018] The backup system 106 also includes a job engine service 220, which is responsible for performing backup and restore operations with a remote agent 280 residing in the client computer 102. The job engine service 220 includes a job manager 230, a media server 240 and a catalog server 250. The job manager 230 is generally configured to receive job requests from the backup server service 210 and to interact with the catalog server 250, the media server 240 and the data server 290 in connection with execution of backup jobs. In addition, during a backup job, the job manager 230 is configured to maintain information regarding volumes that have been completely backed up. This information may be stored as a persistent record of completed volumes 261 in the persistent records memory 260.

[0019] During backup, the media server 240 is generally configured to receive data from the data server 290 and write the data to the storage devices 150. In addition, the media server 240 is configured to maintain information 262 regarding the amount of data (bytes) that have been written to the storage devices 150 and store such information 262 as a persistent record in the persistent records memory 260.

[0020] During backup, the catalog server 250 is generally configured to query and record information about the contents of storage media. More specifically, the catalog server 250 is configured to receive summary information on container and leaf objects of a given volume that is being backed up, and store that information in disk-based catalogs 265 upon successful backup of that volume. Disk-based catalogs 265 may include catalogs in nonvolatile RAM, e.g., MRAM or FeRAM. During the process of backing up a given volume, the catalog server 250 may also be configured to store information regarding the objects being backed up to one or more temporary catalog files 263. Such information may also be stored as a persistent record in the persistent records memory 260.

[0021] As mentioned above, the completed volumes 261, the information 262 regarding the amount of data that have been written to the storage devices 150, and the temporary catalog files 263 may be stored as persistent records. As such, they may be stored in one or more hard drives residing in the backup system 106.

[0022] The remote agent 280 is generally a service that runs on the client computer 102 and allows remote backup and restore of the client computer 102. The remote agent 280 includes a data server 290. During backup, the data server 290 is generally configured to read the data and attributes of container and leaf objects of a selected volume on the client computer 102, and send the data to the media server 240.

[0023] Figure 3 illustrates a flowchart of a method 300 for processing a backup job that was interrupted during a back up process in accordance with one embodiment of the invention. After the backup server service 210 is restarted following a system failure, the backup server service 210 identifies the particular job that was interrupted by the system failure (step 305). The backup server service 210 then resubmits the interrupted job to the job manager 230 for continued backup processing (step 310). In one embodiment, the job may be resubmitted with a flag indicating to the job manager 230 that this job is a "restart" job, not a new job.

[0024] At step 315, the job manager 230 builds a list of volumes associated with the interrupted job and removes from that list a set of volumes that has been completely backed up prior to the system failure to generate a list of volumes that still need to be backed up. The set of volumes that has been completely backed up may be obtained from the persistent record of completed volumes 261.

[0025] At step 320, a determination is made as to whether all the volumes that need to be backed up have been processed. If the answer is in the negative, then the job manager 230 will retrieve a volume from a list of volumes to be backed up and determine whether the retrieved volume is partially backed up (step 323). Generally, only the first volume on the list is partially backed up. If the answer to the query at step 323 is in the negative, then processing continues to step 365, where the job manager 230 initializes the media server 240 to start maintaining a file containing the amount of data that have been written to the storage devices 150. The job manager 230 also initializes the catalog server 250 to start maintaining the temporary catalog files. Subsequent processing following step 365 will be described in detail in later paragraphs.

[0026] Referring back to step 323, if the answer is in the affirmative, then the job manager 230 will use the information 262 regarding the amount of data that have been written to the storage devices 150 and the temporary catalog files 263 to determine whether the retrieved volume from the list of volumes to be backed up was partially backed up prior to the system failure. In one embodiment, the job manager 230 retrieves information regarding the amount of data that have been written to the storage devices 150 (step 325). At step 330, the job manager 230 passes that information to the catalog server 250. At step 335, the catalog server 250 reads the logical block address (LBA) for the last recorded object from the temporary catalog files. LBA is generally defined as the offset of an object from the start of a backup set on the storage devices 150. During backup, in addition to storing objects to temporary catalog files, the catalog server 250 stores LBA for all of the objects in the temporary catalog files 263.

[0027] At step 340, a determination is made as to whether the LBA for the last recorded object is less than the amount of data (bytes) that have been written to the storage devices 150. An answer in the negative indicates that the object has not been written to the storage devices 150. At step 345, the catalog server 250 removes the last recorded object from the temporary catalog files and reads the LBA for the object previous to the last recorded object. Processing then returns to step 340. An answer in the affirmative indicates that all of the objects prior to the recorded object have been written to the storage devices 150. However, the last recorded object may only be partially written and will be backed up again in its entirety in a subsequent backup process. At step 350, the catalog server 250 marks this object as corrupt.

[0028] At step 355, the catalog server 250 uses the objects that are currently listed in the temporary catalog files to create a disk-based catalog containing the partially backed up volume. At step 360, the catalog server 250 passes the information necessary to query this disk-based catalog to the job manager 230. Processing then continues to step 365, where the job manager 230 initializes the media server 240 to start maintaining a file containing the amount of data that has been written to the storage devices 150 and the catalog server 250 to start maintaining the temporary catalog files.

[0029] At step 370, the job manager 230 connects to the data server 290 to begin the process of writing data stored in the client computer 102 to the storage devices 150. At step 375, a volume containing the data from the client computer 102 is retrieved and a determination is made as to whether the volume had been partially backed up prior to the system failure. If the answer is in the negative, then the data server 290 will read the data from the volume (step 376). At step 377, the data server 290 sends the data to the media server 240, which then stores the data to the storage devices 150, and informs the catalog server 290 of the object containing the data to be backed up. The catalog server 290 then stores that information to the temporary catalog files 263.

[0030] At step 378, a determination is made as to whether all the data in the retrieved volume have been stored to the storage devices 150. If the answer is in the negative, then processing continues to step 376. Otherwise, the backup processing of the retrieved volume is complete and processing returns to step 320, where a determination is made as to whether all the volumes that need to be backed up have been processed. Generally, when the backup process of a volume is complete, the job manager 230 performs several wrap up functions, such as notifying the catalog server 250, which then uses the temporary catalog files to generate a permanent disk-based catalog 265 and deletes the temporary catalog files 263, adding a record of the volume to the list of completed volumes, and deinitializing the media server 240, which then deletes its persistent record of data (bytes) written to the storage devices 150. Returning to the query at step 320, if the answer is in the affirmative, which indicates that all the volumes have been backed up, then the job manager 230 will inform the backup server service 210 that the backup job has been completed and delete the list of completed volumes (step 321).

[0031] Referring back to step 375 where a determination is made as to whether the volume retrieved from the client computer 102 had been partially backed up prior to the system failure. If the answer is in the affirmative, then the job manager 230 will send the information necessary to query the disk-based catalog containing the partially backed up volume to the data server 290 (step 379). At step 380, a determination is made as to whether all the container and leaf objects in the retrieved volume have been backed up. If the answer is in the affirmative, then processing returns to step 320, where the determination is made as to whether all the volumes that need to be backed up have been processed.

[0032] Referring back to step 380, if the answer is in the negative, then the data server 290 will retrieve a container object from the retrieved volume and send to the catalog server 250 the information necessary to query the disk-based catalog as to whether the retrieved container object and its contents have been completely backed up, i.e., successfully written to the storage devices 150 (step 385). The conclusions drawn by the catalog queries about which container and leaf objects have or have not been backed up are made possible by the fact that the data server 290 performs an in-order traversal of the volume during backup processing. That is, a depth first search is performed on the volume, and as each container object is encountered during the traversal, that container and all leaf objects contained at that level are backed up before processing of any sub-containers.

[0033] At step 390, the catalog server 250 makes a determination as to whether the retrieved container object is listed in the disk-based catalog containing the partially completed volume. An answer in the negative indicates that the container object and all container and leaf objects below it have not been backed up. No further catalog queries are needed to make this determination, thereby optimizing the network transactions that need to take place. At step 393, the container object and all container and leaf objects below it are backed up and processing then returns to step 380.

[0034] Referring back to step 390, if the answer is in the affirmative, the catalog server 250 will make a determination as to whether another container object exists in the disk-based catalog following the entry for the retrieved container object (step 395). An answer in the negative indicates that while the container object itself was backed up successfully, not all leaf objects below it were backed up. Additionally, it indicates that the container objects below the retrieved container and all of their contents have not been backed up. At step 396, further queries are made regarding each leaf object under the retrieved container, and those not found in the catalog are backed up. Next, all containers objects beneath retrieved container and the leaf objects they contain are backed up. No further queries are needed when backing up these containers and their contents. Processing then returns to step 380.

[0035] Referring back to step 395, an answer in the affirmative indicates that the retrieved container object and all leaf objects contained directly beneath it have been backed up. In this manner, queries need not be made regarding these leaf objects, again optimizing the network transactions that need to take place. Furthermore, if the answer is in the affirmative, the catalog server 250 will make a determination as to whether another container object exists in the disk-based catalogs following the entry for the retrieved container object and with a depth of less than or equal to that of the retrieved container object (step 400). An answer in the negative indicates that while the retrieved container and all leaf objects directly beneath it have been backed up, one or more container objects beneath it and their contents have only been partially backed up. Since these containers will be retrieved and processed by step 380, we need only return processing to step 380.

[0036] Referring back to step 400, an answer in the affirmative indicates that the retrieved container object and all container and leaf objects at all levels beneath it are completely backed up. All of these objects can be skipped without further queries of the disk based catalogs, providing further optimization of the network transactions that need to take place. At step 405, the data server 290 skips the traversal of the container objects under the retrieved container object so that when processing returns to step 380, the next container retrieved will be at a depth less than or equal to the currently retrieved container. Processing then returns to step 380.

[0037] While the foregoing is directed to embodiments of the present invention, other and further embodiments of the invention may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.